DOCUMENT RESUME

ED 304 084 IR 013 602

AUTHOR Stigleman, Sue E.
TITLE The Software Jungle: To Guide or Not To Guide.
PUB DATE May 88
NOTE 8p.; Paper presented at the National Online Meeting (New York, NY, May 10-12, 1988).
PUB TYPE Speeches/Conference Papers (150) -- Reports - Descriptive (141) -- Guides - General (050)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Academic Libraries; *Citations (References); Comparative Analysis; *Computer Software; Computer Software Reviews; *Evaluation Criteria; Feedback; Higher Education; Information Dissemination; *Information Services; Library Services; Microcomputers; *User Needs (Information)
IDENTIFIERS Computer Centers; Software Evaluation; University of North Carolina Chapel Hill

ABSTRACT

This report describes a project which evaluated five IBM and Macintosh bibliographic formatting software programs--ProCite, Sci-Mate, Reference Manager, Notebook II, and Bibliography--and which was conducted by the Health Sciences Library at the University of North Carolina at Chapel Hill at the request of the campus microcomputer support center. Descriptions of project activities and how well they worked are presented in four categories: selecting and communicating criteria for evaluation; knowing users' needs (expanding your perspective beyond the library); working with software producers; and establishing responsibility and limits for supporting the chosen software. Reactions to the evaluation project and the selected programs are reported, and nine steps are suggested for anyone undertaking a similar project: (1) publicize the fact that the evaluation is being initiated; (2) actively solicit user input; (3) arrange software presentations by users and vendors; (4) visit a few active users of some of the programs; (5) develop and articulate the evaluation criteria; (6) conduct structured evaluations of all of the software being considered; (7) make the selection and advertise it along with the evaluation criteria and a description of the evaluation process; (8) solicit feedback from users on an ongoing basis; and (9) continue to monitor new versions and new programs, comparing them to your original choice(s) and user feedback. (CGD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED304084

The Software Jungle: To Guide or Not to Guide

Sue E. Stigleman
Health Sciences Library
CB# 7585
University of North Carolina
Chapel Hill, NC 27599

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Sue E. Stigleman
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Keywords: Bibliography Formatting Software, Evaluation.

Abstract: A rapidly developing segment of the software market is producing a growing number of information management products. These products perform such tasks as formatting bibliographic citations, searching national online databases, and creating and searching personal databases, both bibliographic and full text. Because of the obvious similarities between these software programs and a number of typical library services, such as online searching and cataloging, library staff are often asked to recommend or choose software for library users.

At UNC-CH, the Health Sciences Library was asked to recommend bibliography formatting software programs as candidates for official support by the campus Microcomputing Support Center. After evaluating the existing programs, two were recommended and are now being supported on a university-wide basis.

The process raised a number of issues which people encountering similar requests may want to consider, including 1) selecting and communicating criteria for evaluation; 2) knowing users' needs: expanding your perspective beyond the library; 3) working with software producers; 4) establishing responsibility and limits for supporting the chosen software.

1. INTRODUCTION

In common with many libraries, we at the Health Sciences Library have kept a particularly close eye on two types of software: end-user searching software and bibliography formatting software. We have offered workshops in end-user searching and reprint file management for several years, and the microcomputer component of these workshops has become increasingly important. In 1986, I began taking a thorough look at the existing bibliographic programs with the intention of settling, at least for the next few years, on a single program to use in the reprint filing workshops. Concurrently, the library was having discussions with the campus Microcomputing Support Center about various issues involved in providing campus-wide user support for this type of software. The Microcomputing Support Center was interested in providing support, but wanted to narrow down the field rather than trying to support all of the

IR013602

programs. We were asked to recommend two programs, at least one of which should be relatively inexpensive for student use.

I would like to share observations and suggestions about the process of recommending software for widespread use. I know from experience that it's easy to overlook some things that will result in a successful recommendation. For each of four areas, I will describe what we did and how well it worked. I will finish with some overall observations on the evaluation project and with a recommended set of steps to use.

2. SELECTING AND COMMUNICATING CRITERIA FOR EVALUATION

A whole talk could be given just on criteria for evaluating bibliographic software. Briefly, the criteria I used were:

- Hardware used: The Health Sciences Library and Microcomputing Support Center were interested only in software for the IBM and Macintosh families of microcomputers, so we ignored the other main families, such as the non-Macintosh Apples.

- Reasonably large size limitations for the maximum number of fields per record, maximum number of records per file, and the maximum characters allowed for the various elements of a bibliographic record. One program was eliminated because it allowed a maximum of 10 unique keywords in the entire file.

- Flexibility in document types, including support for more than just journal articles and books.

- Full-screen data entry and editing.

- Reasonably powerful searching features, such as Boolean operators, truncation, nesting, field qualification, and proximity searching.

- Flexible sorting, including multiple levels and user selected fields.

- Support for several citation styles and the ability to define additional formats.

- Ability to scan a word processor manuscript and automatically generate a bibliography of the references cited in the manuscript.

- The ability to import and export data. People may later need to switch to a different program.

- The ability to import downloaded online search results, since, among other considerations, this is the most efficient means of data entry.

- Ease of use.
- Vendor reliability and support.
- Price.

After evaluation, the major contenders for the high end programs were Pro-Cite, Sci-Mate, and Reference Manager. Notebook II, along with its companion program, Bibliography, was the only serious low-end contender. After considering all of the factors, I recommended Pro-Cite and Notebook II/Bibliography. The Microcomputing Support Center also later added support for Reference Manager.

None of the programs are perfect, all of them possessing some major limitations. As a result, I had to balance the programs' various strengths and weaknesses against each other. For a given user, the criteria may be different, and almost certainly the relative weight of the different criteria will differ.

I particularly tried to avoid what I felt was likely to be a built-in bias towards the health sciences part of our campus, a bias due to 4 factors:

(1) I work in a medical library;
(2) a great deal of the interest in bibliographic software on our campus was in health sciences;
(3) almost all of the end-user searching at UNC appeared to be done in health sciences;
(4) the most commonly searched database on campus is MEDLINE.

I felt since I was evaluating for general campus-wide support, I should try to recommend software usable by a large number of diverse types of people.

Regardless of the criteria used or the weights assigned to them, a difficult task in software selection is communicating criteria for evaluation so that users will understand them. I've observed that this is true for all software, but it's very noticeable for bibliographic software. Many people have never actually used this type of software before, and consequently have a very limited view of what the software can do or what they want it to do. Even people who have used the software won't always understand the criteria you use for the selection. The training coordinator at the Microcomputing Support Center observed: "In our attempt to locate trainers, I found that most users have a fragmented idea of a given package's capabilities. While they tend to be very competent in presenting aspects used with their projects, they are often wholly unaware of other features."

How well did the criteria work?

One faculty member disagreed vigorously with my recommendations. His comments illustrate clearly the difficulties in weighing the criteria when none of the programs fully satisfies them. A longtime user of Reference Manager, he believed that Reference Manager should have been chosen over Pro-Cite. He argued that the single most important task that this type of software performs is the one of scanning a word processor

manuscript and automatically generating a bibliography of the references cited. In his view, the program that does this one function the best is the one that should be chosen. He is right that Reference Manager does do this task better than Pro-Cite. I disagreed with his assumption that his needs are the same as those of all of the potential users of the software.

The faculty member also downplayed the idea that anyone would actually use document formats other than journal articles and books, and he felt that my viewpoint represented some kind of idiosyncratic view of what libraries want for their own use, rather than genuine user needs. The philosophy underlying these comments is that a software choice should be based on the most common user profile. My philosophy was also to be concerned about the user who has more unusual needs, a factor which ended up favoring Pro-Cite over Reference Manager. Either of these viewpoints can be used -- the choice will vary in different situations.

3. KNOWING USERS' NEEDS

Because of my familiarity with bibliographic programs and with some of the tasks this software performs, such as searching and formatting citations, I relied heavily on my own expertise in developing the evaluation criteria. I used conversations I had had with people who had come to the library over the past few years to talk about their needs and/or their experiences with the programs. I also drew on the work of people in other libraries, most notably Abigail Hubbard at the Academy of Medicine Texas Medical Center and Joe Wible at Stanford. There were relatively few existing users of the software on campus, and no identified experts. Neither the other campus libraries nor the Microcomputing Support Center had become involved in using or evaluating the software.

How well did this approach work? It worked well for the actual evaluation itself. My background is broad enough that I feel I took a thorough and objective look at the programs. The difficulties, as I have said before, came with the problems in weighting the different criteria. Although I had a fairly good idea of the likely user needs, more actively soliciting user input would have helped in two ways.

First, it would have had a psychological effect in assuring people that their viewpoint was being considered in the evaluation process.

Second, it would have given me a broader profile of the current and potential users, which would have been useful in weighting the criteria.

In preparing this paper, I asked for feedback from various people who had been involved with the evaluation process. One of the people from the Microcomputing Support Center observed that evaluations can be obtained from magazines. She felt that three things needed to be tied together for a successful evalution: objective evaluations, reactions from current users of the software, and considerations of the specific applications in use or planned on our campus.

4. WORKING WITH SOFTWARE PRODUCERS

In this evaluation project, we didn't work with the producers in a systematic way. Because we were using Pro-Cite in the library, I had frequent conversations with the company and knew about their plans for the next version. I didn't have that level of knowledge of the plans for other software programs.

If I were doing the evaluations again, I would contact the companies of the most likely candidates and ask them for their plans for their next release, including both features and timetable for release (checking with current users of the programs to find out how reliable the information from the company usually is.) I would also provide the producers with the results of my evaluation and ask them to address the areas that I perceived to be their programs' major weaknesses.

How much should you base your selection on the programs as they exist at that moment, and how much should you find out or rely on plans for the future? If a new version that corrects a major flaw is being released in 2 months, I would consider it strongly. However, it would be doing a grave disservice to future software users to base a decision on supposed improvements that turn out to be "vaporversion" -- an upgrade talked about but never actually released.

Another note: I didn't negotiate the volume purchasing agreements, but my sense is that the competitiveness created by looking seriously at more than one package helps reduce the prices offered by the producers.

5. ESTABLISHING RESPONSIBILITY AND LIMITS FOR SUPPORTING THE CHOSEN SOFTWARE

What happens after the recommendation? At UNC, the Health Sciences Library didn't have the resources to offer day-to-day support for the programs. In a series of meetings including people from the Library and the Microcomputing Support Center, we negotiated a division of labor.

The Microcomputing Support Center:
- made the final selection of programs to support;
- supports the chosen software, including negotiating site licenses and volume purchase agreements, assisting users in choosing a program, providing walk-in or phone-in user support services, and teaching the use of the programs;
- negotiates volume purchase agreements or site licenses;
- creates and collects citation styles for public distribution (not yet being done);
- provides evaluation copies of bibliographic software programs through its software library.

The Health Sciences Library:
- taught the first campus-wide short course;

taught a training session for Microcomputing Support Center user support staff;
evaluates new versions of all of the bibliographic programs and makes evaluations available through the Microcomputing Support Center;
provides specific applications workshops using the software (not yet being done).

How well has this worked? The Microcomputing Support Center's experiences with user support have been tough. The Microcomputing Support Center trainers and staff don't use any of the programs in their daily work. The number of questions from users haven't been frequent enough to really get the user support staff trained. The staff must refer to the manual for every question. Some calls get referred to the library for resolution.

The division of responsibility has generally worked out well for the Library, although we have been unable to offer the workshops due to staffing shortages.

6. OVERALL OBSERVATIONS

The Microcomputing Support Center has been very pleased with the reactions to the evaluation project. They didn't have the time or the expertise to do the evaluation and appreciated the Library's willingness to step in and help them make a decision about which programs to support. Their one suggestion for improvement was to be more aggressive in soliciting user input.

Reactions to the selected programs overall are positive. People seem pleased at having support for two well-known programs, each with a base of users. Reference Manager and Pro-Cite are both popular, with Reference Manager generally being used more in the health sciences and Pro-Cite used more in Academic Affairs.

Notebook, the low-end program we hoped would be used by students, hasn't proven popular. Student use is low, possibly because students don't want to learn another piece of software in order to do their theses or dissertations. It may still cost too much for students. We may need to lower our standards in order to find a limited but cheap program as a starter package.

Reactions to bibliographic software in general can be summed up in this quote from the director of training at the Microcomputing Support Center: "Users seem largely unaware of how to apply these packages to their bibliography needs and are also relatively ignorant of the effort required to use them effectively. It is tedious and expensive to utilize the full power of the software."

7. RECOMMENDED STEPS

Finally, to summarize, here are a series of suggested steps to consider using if you undertake a similar evaluation process.

- Publicize the fact that the evaluation is being initiated. Both the Microcomputing Support Center and I neglected to do this, with the result that the evaluation came as a surprise and was perceived to be a fait accompli.

- Actively solicit input from users. I suggest an open meeting for people interested in the type of software to discuss what applications they have in mind, what program(s) they are already using, and how well the programs work.

- Arrange software presentation by users and vendors, trying to balance the perspectives.

- Visit a few active users of some of the programs. This is not solely to get reactions to a specific program, but also to see how they use the type of software in their work.
- Develop and articulate the evaluation criteria.

- Conduct structured evaluations of all of the software being considered. Document the results.

- Make the selection and advertise it along with the evaluation criteria and a description of the evaluation process.

- Solicit feedback from users on an ongoing basis.

- Continue to monitor new versions and new programs, comparing them to your original choice(s) and user feedback.